



The Information Interface

PRESS
RELEASE



The Modern Integrated Product Solution for

ZIM THE INFORMATION INTERFACE

ZIM is the first Information Management system to take the current and future generations of general-purpose microcomputers seriously. **ZIM** is the first product to take advantage of the most recent and significant developments in modern data base research — in particular the highly acclaimed Entity-Relationship Model of data (see diagram). **ZIM** is the first product to address the dual requirements of Information Management — Maintenance and Organization.

ZIM was designed and built in the current microcomputer environment but has functional and architectural characteristics exceeding the most modern of mainframe systems. **ZIM** is a totally

integrated product incorporating the historically separate functions of data dictionary, data maintenance language, query language, report writer and forms facilities in a uniform and elegant manner. **ZIM** is a complete 4th Generation Applications Development Language which may be compiled for maximum production performance. Under operating systems which support the facility, **ZIM** can execute in full multi-user mode.

ZIM is aptly named The Information Interface. For the rapid development of the complete spectrum of information management applications ranging from the simple to the most complex, **ZIM** is the only product required.

ZIM AT A GLANCE

ADVANCED

- Uses the recent Entity-Relationship Model of data.
- Has a powerful English-like command language.
- A fully integrated system — not a collection of utility programs.
- Has its own internal Hierarchical Directory facility.
- Has a fully integrated Data Dictionary.
- Is a complete 4th Generation Applications Development language.
- Can operate in full multi-user mode.

POWERFUL

- The expressive capability of **ZIM**'s command language is unsurpassed.
- A full function user-designed forms facility.
- A full function report writer.
- Full-screen browse/update of any data base subset.
- Wide range of field types.
- Full set of built-in functions.
- Easily handles complex Relational theory operations not even supported by most Relational systems (e.g. the outer join).
- Excels in bulk operations on multiple members from multiple tables.

EASY TO USE

- The Entity-Relationship Model is the most natural approach to data base design and implementation.
- **ZIM**'s command language reads like plain English — not algebra.
- The Hierarchical Directory facility provides for natural information organization.
- Simple concepts apply uniformly across all functions.
- **ZIM**'s command language makes intelligent assumptions thus avoiding needless detailed specifications.
- Operating system editor invoked from within **ZIM**.
- Forms designed by drawing them on the screen.
- Includes an interactive tutorial for fast familiarity.

FLEXIBLE

- Multiple related or unrelated data bases within one environment.
- Ad hoc queries and reports are easy.
- Simple user commands for more common requests.
- Rapid development of complete forms-based applications with sophisticated reports.
- Provides editing, querying, loading from, and unloading to, standard operating system files.
- Device independent I/O re-direction (console, printer, file).
- Programming Language Interface for special applications.

EFFICIENT

- Swift command parsing.
- Compilation of user commands for maximum performance.
- Priority driven, highly buffered I/O.
- B*-tree indexes provide fast retrieval and updating.
- Optimal data access strategy automatically selected.
- Ultra fast sorting on multiple fields.

ELEGANT

- Full integration means single environment for all information management operations.
- The Entity-Relationship Model is the basis for the most natural approach to planning and designing data bases.
- A natural English-like command language.
- The Directory facility is the basis for natural information organization.
- Common simple concepts of information selection apply uniformly across all functions.

all Information Management Requirements

ZIM AND THE E-R MODEL

ZIM completely understands a data base designed using the E-R model. **ZIM** imposes no limit on the number of entity sets, relationships or roles which may be defined. Relationships may be of any degree (binary, ternary etc.), may have data fields associated with them, and may be defined between entity sets in different **ZIM** directories. **Reflexive** relationships (e.g. parts **Contain** parts) may be defined to easily support Bill of Materials (BOM) structures.

ZIM'S COMMAND LANGUAGE

The Entity-Relationship model provides the basis for **ZIM**'s elegant, English-like, and exceptionally powerful command language. Consider a typical Relational data base language command for the query "list all the employees who work on project Alpha and display all fields".

Typical Relational Language Command

```
select * —  
from workontab,projects,employees —  
where workontab.enum=employees.enum —  
and workontab.pnum=projects.pnum —  
and projname='Alpha'
```

Equivalent ZIM Command

list all employees workon projects where projname=Alpha

It can be seen that the structure of **ZIM**'s command language reads like plain English. This command simplicity applies to all functions within the **ZIM** environment yet it is extremely powerful; capable of handling the most difficult problems in modern data base theory with ease. For example, the vast majority of Relational systems can only handle the situation where entries in several tables are related. Of equal importance is the ability to retrieve table entries which are not related or a combination of both (for readers familiar with Relational data base theory, this corresponds to the outer join problem). **ZIM** reduces this complex issue (and several others) to simple and meaningful keyword modifiers to the normal command. For example, to list all the employees who do not work on any projects, the **ZIM** command would be

list all employees(unrelated) workon projects

To achieve this simple result in most popular relational languages takes many complex commands — if it can be achieved at all.

ZIM INFORMATION MANAGEMENT FEATURES

INTEGRATION

ZIM is not a collection of separate utility programs but rather, a single integrated system. Once in the **ZIM** environment, all facilities are available immediately. Common concepts and command syntax apply uniformly to all functions.

Information Management may be broken down into a number of sub-functions. Shown below are the **ZIM** features available under each sub-function — all within the **ZIM** system environment.

INFORMATION REPRESENTATION

- Natural and flexible data base planning and design using the Entity-Relationship Model of data.
- Any number of Directories, Entity Sets, Relationships, Roles.
- Relationships may be of any degree (binary, ternary etc.) and may have fields defined in them.
- Role names available for even greater semantic clarity.
- Any number of fields in an Entity Set or Relationship.
- Wide range of field types.
- Any field may be indexed. Both multi-valued and unique-valued indexes are supported.
- Definition data is entered into a pre-defined XIM data base, in exactly the same manner as user data is entered and updated. This provides a powerful and naturally integrated Data Dictionary.

INFORMATION ORGANIZATION

- **ZIM**'s Directory facility allows the appropriate partitioning of complete applications into functional data bases (e.g. Personnel, Project Control).
- Relationships may be defined between Entity Sets in different directories.
- Multiple directories may be accessed simultaneously.
- Definitions may be copied or transferred from one directory to another.

INFORMATION MAINTENANCE

- Only three commands required for all update operations, ADD, CHANGE and DELETE. The flexibility of these commands is unparalleled.
- Source data for update operations may be derived from explicit assignments from constants, variables, or general expressions; input from the console; from formatted or free-form operating system sequential files; from a specified subset of the existing data in the data base; or from forms
- All indexes automatically updated.
- Extraction of any subset of the data base to operating system sequential files (for use by other systems).
- The user-defined forms facility provides the most natural interface for update operations.



1785 Woodward Drive, Ottawa, Ontario K2C 0R1 (613) 727-1397

Press Release

Press Release

INFORMATION RETRIEVAL

- In **ZIM**, information to be retrieved is stated in the form of a set specification such as
employees workin departments where salary > 20000
A set specification may include any number of entity sets, relationships and roles, an arbitrary selection (WHERE) clause, and set operations with other set specifications. The command that precedes a set specification determines the action to be taken.
- All set specifications are automatically analyzed to select the most efficient data access strategy.
- If a set specification is preceded by the LIST command as in
list employees workin departments where salary > 20000
then the retrieved information is directed to the selected output device (console, printer, operating system file, etc.)
- If a set specification is preceded by the FIND command as in
find employees workon projects where projname =Alpha
then **ZIM** simply remembers where the information which satisfies the set specification is. This remembered information is called a "set". Sets may be browsed in a line-by-line or full-screen mode and data base updates may be made through them. Once created, sets may be used in subsequent retrieval operations.
- If a set specification is preceded by the REPORT command as in
report from employees workin departments sorted by deptnum
then the report writer is invoked.

INFORMATION PRESENTATION

- The **LIST** command provides the familiar tabular presentation of data with computed fields and simple formatting options.
- The **REPORT** command is capable of generating the most complex of detailed and summary type reports with report headings and footings, page headings and footings, multiple control breaks with headings and footings, intelligent handling of long text fields, conditional control etc. The report writer also has a default mode in which simple ad hoc reports require the minimum of specifications.
- **ZIM**'s user-designed forms facility allows forms to be "drawn" on the screen. In use, multiple forms and multiple occurrences of each form may be displayed at the same time. The forms facility provides full data editing with intelligent treatment of long text fields.
- **ZIM** provides a full range of built-in functions for operating on data to be presented.

INFORMATION SHARING

- Under operating systems which support the facility, **ZIM** can execute in full multi-user mode.
- Control over information sharing/privacy is accomplished through **ZIM**'s unique directory facilities.

ZIM APPLICATION DEVELOPMENT FEATURES

4TH. GENERATION LANGUAGE

ZIM's natural command language syntax makes ad hoc retrieval or update operations easy. For repetitive operations, **ZIM** provides the facility for creating new commands — user commands. Structured programming constructs are provided for execution flow control. User commands form the basis for **ZIM**'s exceptionally powerful 4th. Generation Application Development facilities. **ZIM** also provides the ability to **compile** user commands for maximum performance of production applications.

USER COMMANDS

- Are simple operating system files containing **ZIM** commands.
- All **ZIM** functions may be invoked from within user commands.
- Are created and subsequently modified by the **ZIM** EDIT command.
- Parameter values may be passed to them.
- May call other user commands (including themselves!)

VARIABLES AND FUNCTIONS:

- Variables of all types.
- Full range of functions for all field types.

STRUCTURED PROGRAMMING

- IF...ELSE...ENDIF
- WHILE...ENDWHILE
- INPUT
- OUTPUT
- LET (assignment)

COMPILATION

- Complete applications comprising many user commands may be compiled.
- Eliminates the requirement for command parsing, strategy selection and execution code generation.
- Reduces run-time memory requirements.
- Delivers maximum application performance.

PROGRAMMING LANGUAGE INTERFACE

- Only required for special function applications (e.g. real-time process control systems).
- All of **ZIM**'s power is available to the host program.



INFORMATION RETRIEVAL

- In **ZIM**, information to be retrieved is stated in the form of a set specification such as
employees workin departments where salary > 20000
A set specification may include any number of entity sets, relationships and roles, an arbitrary selection (WHERE) clause, and set operations with other set specifications. The command that precedes a set specification determines the action to be taken.
- All set specifications are automatically analyzed to select the most efficient data access strategy.
- If a set specification is preceded by the LIST command as in
list employees workin departments where salary > 20000
then the retrieved information is directed to the selected output device (console, printer, operating system file, etc.)
- If a set specification is preceded by the FIND command as in
find employees workon projects where projname =Alpha
then **ZIM** simply remembers where the information which satisfies the set specification is. This remembered information is called a "set". Sets may be browsed in a line-by-line or full-screen mode and data base updates may be made through them. Once created, sets may be used in subsequent retrieval operations.
- If a set specification is preceded by the REPORT command as in
report from employees workin departments sorted by deptnum
then the report writer is invoked.

INFORMATION PRESENTATION

- The **LIST** command provides the familiar tabular presentation of data with computed fields and simple formatting options.
- The **REPORT** command is capable of generating the most complex of detailed and summary type reports with report headings and footings, page headings and footings, multiple control breaks with headings and footings, intelligent handling of long text fields, conditional control etc. The report writer also has a default mode in which simple ad hoc reports require the minimum of specifications.
- **ZIM**'s user-designed forms facility allows forms to be "drawn" on the screen. In use, multiple forms and multiple occurrences of each form may be displayed at the same time. The forms facility provides full data editing with intelligent treatment of long text fields.
- **ZIM** provides a full range of built-in functions for operating on data to be presented.

INFORMATION SHARING

- Under operating systems which support the facility, **ZIM** can execute in full multi-user mode.
- Control over information sharing/privacy is accomplished through **ZIM**'s unique directory facilities.

ZIM

APPLICATION DEVELOPMENT FEATURES

4TH. GENERATION LANGUAGE

ZIM's natural command language syntax makes ad hoc retrieval or update operations easy. For repetitive operations, **ZIM** provides the facility for creating new commands — user commands. Structured programming constructs are provided for execution flow control. User commands form the basis for **ZIM**'s exceptionally powerful 4th. Generation Application Development facilities. **ZIM** also provides the ability to **compile** user commands for maximum performance of production applications.

USER COMMANDS

- Are simple operating system files containing **ZIM** commands.
- All **ZIM** functions may be invoked from within user commands.
- Are created and subsequently modified by the **ZIM** EDIT command.
- Parameter values may be passed to them.
- May call other user commands (including themselves!)

VARIABLES AND FUNCTIONS:

- Variables of all types.
- Full range of functions for all field types.

STRUCTURED PROGRAMMING

- IF...ELSE...ENDIF
- WHILE...ENDWHILE
- INPUT
- OUTPUT
- LET (assignment)

COMPILATION

- Complete applications comprising many user commands may be compiled.
- Eliminates the requirement for command parsing, strategy selection and execution code generation.
- Reduces run-time memory requirements.
- Delivers maximum application performance.

PROGRAMMING LANGUAGE INTERFACE

- Only required for special function applications (e.g. real-time process control systems).
- All of **ZIM**'s power is available to the host program.

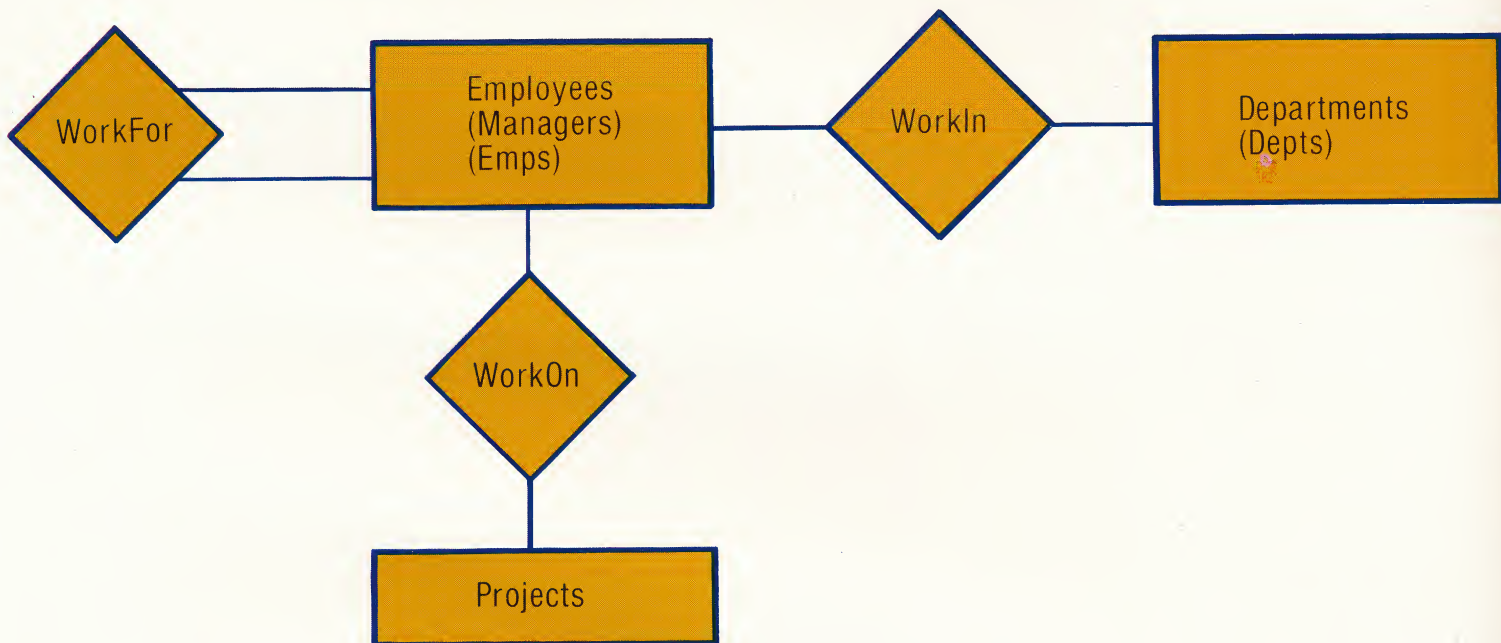
all Information Management Requirements

THE ENTITY-RELATIONSHIP MODEL OF DATA

The Entity-Relationship Model of data (known as the E-R Model) is the most significant recent development in modern data base research since the proposal (in 1970) of the Relational Model of data. The E-R model extends the Relational model to include (amongst other things), defined relationships between the tables in a data base. This greatly enhances the ability to think about data base structures, as the relationships between tables are as important as the tables themselves. For example, Employees **Work In** Departments; Employees **Work On** Projects; Parts **Contain** Parts; Libraries **Issue** Books to People etc. The E-R model has been universally acclaimed as the most natural approach to designing data bases.

Tables are called **Entity Sets** in the E-R model as they represent sets of things (entities) which have the same characteristics. Consider a data base consisting of the three entity sets **Employees**, **Departments** and **Projects**. The following diagram is an Entity-Relationship **Schema** for such a data base. It could be sketched out in a few seconds. In this schema, the rectangles represent **entity sets** and the diamonds represent **relationships**. Names shown in brackets are **role** (alternative) names. It is clear that the schema shows not only the things we wish to work with, but also, how they relate to each other in the context of the application being planned.

An Example E-R Data Base Schema



From schema diagrams such as this, the functions that a data base is intended to address become immediately apparent. There is no directionality associated with a relationship. For example, the relationship **WorkIn** could be used to find either, the Employees who work in a specified set of Departments or, conversely, the Department information for a specified set of Employees.

To further clarify the design and use of a **ZIM** E-R data base, **role names** may be defined. For example, **Managers** has been defined as a role name for Employees. The relationship **WorkFor** has been defined between Employees and Managers (i.e. Employees **WorkFor** Managers). The role name **Employ** could be defined for the relationship **WorkIn** to clarify the fact that Departments **Employ** Employees as well as Employees **WorkIn** Departments.

ZIM

PERFORMANCE FEATURES

ZIM delivers impressive performance on the current generation of micro-computers primarily due to the compact Executor module which performs the data base access operations. The architecture of **ZIM** is such that it will be upwards compatible with advances in data base research and technology. Some of the other features which contribute to **ZIM**'s performance are listed below.

INDEXES

- Indexes are provided through a new and very compact implementation of B⁺-trees. Access time is very fast and independent of the distribution of data values in a field.

SORTING

- Most sort requests will execute in memory.
- Only the specified sort fields data is manipulated. Thus much more sort data can be kept in memory.

OPTIMAL STRATEGY SELECTION

- The Strategy Selector module of **ZIM** automatically analyzes an information access request (a set specification) on any command and selects the optimal data access techniques. Thus the requested information is delivered in the most efficient manner possible.

HIGHLY BUFFERED I/O

- A priority driven buffering scheme is employed to ensure that the most frequently accessed information is maintained in memory.
- The Data Dictionary information crucial to command parsing is maintained in memory.



1785 Woodward Drive,
Ottawa, Ontario K2C 0R1
(613) 727-1397

Press Release

Press Release

FOR RELEASE AFTER NOVEMBER 14, 1984

For further information contact: **Steve Denison**
Director, Marketing
Zanthe Information Inc.
OTTAWA, Ont.
Tel: (613) 727-1397

Zanthe Introduces Full Multi-User ZIM Under XENIX

OTTAWA, Ontario -- Zanthe Information Inc. announces that ZIM®, the first Information Management system to implement the Entity Relationship Model of Data, now can run under XENIX®, a version of UNIX® based on the Motorola 68000.

The XENIX implementation is fully compatible with all other ZIMs. ZIM also runs under PC-DOS, MS-DOS, and under QNX. ZIM under XENIX also fully supports TERMCAP for different types of terminals.

"With the latest enhancements Zanthe Information has made in ZIM, coupled with the enhanced multi-user support system we have developed, there is little doubt that ZIM is the industry leader in implementing the Entity Relationship (ER) model of data," said Chris Turnbull, Vice-President, Research and Development of Zanthe Information.

"The importance of ZIM operating under the different environments of XENIX, PC-DOS, MS-DOS, and QNX is that it permits any machine that supports those operating systems to draw on the enormous versatility of ZIM," declared Turnbull.

For multi-user applications, ZIM has a number of key features that enhance performance. Among those are page locking, automatic transactions, and user-defined transactions.

(OVER)

Page locking is of premium use in a multi-user complex, ensuring that work underway can be completed by an individual without having the information being altered simultaneously by someone else who wishes to work on the same section.

The automatic transaction feature helps speed up production by obviating other keystroking. User defined transactions, with features such as abort or commit, also make for ease of use, besides speeding up operations.

ZIM is available on IBM PC®, XT® and AT®, Monroe 2000®, Columbia®, Compaq®, Corona®, Digital®, Eagle®, Hewlett Packard®, Tandy 2000®, and many other Personal Computers as well as 68000-based systems supporting XENIX.

November 1984

Zanthe Information Manager

	<u>Suggested Retail Price List (\$US)</u>	<u>Qty</u>	<u>Discount*</u>	<u>Price</u>
ZIM	System Diskettes, Interactive	1	-	\$795.00
	Tutorial and Full Documentation	2 - 5	10%	715.00
	(MS-DOS)	6 - 25	20%	636.00
		26 - 50	28%	570.00
		Over 50	Negotiable	
	System Diskettes, Interactive			
	Tutorial and Full Documentation			
	(QNX)			\$975.00
	Run time Package for Independent			Available
	Software Developers			on request

OPTIONS

	QNX Multi-User Package	\$500.00
	Programming Language Interface	\$125.00
	Compiler	\$125.00
ZIM EVALUATION	Interactive Tutorial and Tutorial Documentation	\$25.00

Dealer, Distributor and University Inquiries welcome.
Prices are subject to change without notice. MS-DOS is a
Micro Soft Corporation trademark, QNX is a Quantum Software
Systems trademark.

* Discount schedule applies to all product prices

ZIM SPECIFICATIONS

ZIM is a fully integrated data base management, application development system that implements the Entity Relationship ("ER") model of data, and as such is a significant improvement over relational data bases. The ER model represents a breakthrough in data base technology providing the user with more power, speed and ease of use than anything that has gone before it, and ZIM is the only system that has it.

ZIM REQUIREMENTS

ZIM runs under PC-DOS, MS-DOS (version 2.0 or later) and under QNX (version 1.15 or later), a multi-user, multi-tasking, networking environment. ZIM will also be available in the very near term under XENIX, and UNIX.

ZIM is available on most of the machines that support the operating systems described above. They include amongst others: IBM PC, XT and AT, Eagle, Compaq, Hyperion, Columbia, Corona, Monroe, GEAC, Tandy, Hewlett Packard, and Digital.

ZIM requires a minimum of two floppy disks or one floppy disk and a hard disk. Under DOS the system requires 320K, under QNX the system requires 384K of memory. The ZIM evaluation system requires only 256K.

The ZIM Programming Language Interface requires the Lattice - C compiler system under DOS or the Quantum C compiler for QNX.

SYSTEM LIMITATIONS

The following is a description of the limitations imposed on the user of ZIM. In this instance, "NO LIMIT" means the system is limited only by hardware resources.

i)	Number of files per data base	NO LIMIT
ii)	Number of record types per data base	NO LIMIT
iii)	Number of records per file	NO LIMIT
iv)	Maximum file size	256MB (DOS)
v)	Number of characters per record	8192
vi)	Number of fields per record	NO LIMIT
vii)	Number of relationships per data base	NO LIMIT
viii)	Number of fields a sort can be executed on	NO LIMIT
ix)	Size of data base	NO LIMIT
x)	Number of users	NO LIMIT
xi)	Number of data bases	NO LIMIT



PRESS RELEASE

PRESS RELEASE

FOR RELEASE AFTER NOVEMBER 14, 1984

For further information contact:

**Steve Denison
Director, Marketing
Zanthe Information Inc.
OTTAWA, Ont.
Tel: (613) 727-1397**

ENHANCEMENT OF ZIM ANNOUNCED

OTTAWA, Ontario -- Zanthe Information Inc. has enhanced the capabilities of ZIM, the first commercially available Data Base Management System to implement the Entity Relationship (E-R) Model of Data, making it even more versatile for the end user than any other PC-related software. It remains the only PC product to efficiently address the dual requirements of Information Organization **and** Management.

"ZIM 2.4 provides the most complete applications development language and environment for database design in such a logical method that even the most inexperienced computer user can quickly translate information problems into clear database solutions," said Alan Price, Zanthe's president.

ZIM 2.4 integrates new database facilities, many of them unique to ZIM, into a single system with a common syntax for all functions. Among the new forms features are:

- | | |
|---|---|
| ☐ Extended defaults; | ☐ On-screen numeric and |
| ☐ Setable transmit keys; | alphanumeric validation; |
| ☐ Required fields; | ☐ Automatic transmit fields; |
| ☐ Automatic skip fields; | ☐ Automatic clear fields; |
| ☐ Foreground color; | ☐ Background color; |
| ☐ Fill characters; | ☐ Upper, lower, or mixed case; |
| ☐ Multiple display | ☐ Left, right, centered or |
| attributes; | no justification. |

(OVER)

In addition, form field attributes may be dynamically changed at any time to suit the varying needs of an application.

ZIM also supports variable length fields and records. In all, there are 50 new built in functions including extended mathematical and extended alphanumeric manipulation capabilities. Several extensions have been made to the report writer facilities. Additional features include:

- ☐ Optional compiler for maximum performance;
- ☐ Sample applications on the distribution disks;
- ☐ Form driven novice user interface for building a database.

The new ZIM features are added to those of the original ZIM software package that included:

- ☐ Data dictionary;
- ☐ Forms "painting";
- ☐ Comprehensive report writer;
- ☐ Hierarchical Directory facility.
- ☐ Powerful query/update language;
- ☐ Flexible forms manipulation;
- ☐ Complete Fourth Generation Applications Development Language;

Applications may be developed in ZIM up to 20 times faster than with traditional methods. On operating systems that support the facility, ZIM executes in multi-user mode. Added ZIM strengths are the fact that it is one of the fastest database systems available today. On an IBM PC-XT it can sort 1,000 records in 35 seconds and 10,000 in 10 minutes. Indexed retrieval of any one of more than 40,000 records takes an average of four seconds.

In fact, the limits of ZIM's versatility depends solely on the underlying operating system.

ZIM is available under MS DOS and Quantum Software Systems' QNX. It is supported for a wide variety of machines including IBM PC®, XT® and AT®, Monroe 2000®, Columbia®, Compaq®, Corona®, Digital®, Eagle®, Hewlett Packard®, Tandy 2000®, and many other Personal Computers as well as 68000-based systems supporting XENIX.